

2176 2776
Docket No. #3
END920000080UST(13679)

TRANSMITTAL OF FORMAL DRAWINGS

In Re Application Of: **James Richard Wason**

Serial No.	Filing Date	Batch No.	Examiner	Art Unit
09/616,809	July 14, 2000	Unknown	Unknown	Unknown

Invention: **TEXT FILE INTERFACE SUPPORT IN AN OBJECT ORIENTED APPLICATION**

RECEIVED

Address to:
**Assistant Commissioner for Patents
Washington, D.C. 20231**

**JAN 18 2001
Technology Center 2100**

Transmitted herewith are:

3 sheets of formal drawing(s) for this application.

Each sheet of drawing indicates the identifying indicia suggested in 37 CFR Section 1.84(c) on the reverse side of the drawing.

John S. Sensny
Signature

**John S. Sensny
Registration No. 28,757**

**SCULLY, SCOTT, MURPHY & PRESSER
400 Garden City Plaza
Garden City, New York 11530
(516)742-4343**

Dated: **January 5, 2001**

I certify that this document and attached formal drawings are being deposited on January 5 2001 with the U.S. Postal Service as first class mail under 37 C.F.R. 1.8 and addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.
<i>Janet Giordano</i> Signature of Person Mailing Correspondence
Janet Giordano Typed or Printed Name of Person Mailing Correspondence

Processing a Text File in a Computer Application

Forming a template from
fragments of the text file.

Using the template as an overlay
for parsing incoming files, or as
a prototype to generate a segment
of an output file.

Using a macro class to map data
from the text file to an application.

Embedding the macro class as a
keyword within the template, wherein
when the template reaches the
keyword, the template calls the macro
class to further process the text file.

Figure 1

1. Templates that support variable substitution and conditional or iterative generation for output files.
2. A mechanism to fill in variable substitutions in the templates.
3. A mechanism to handle iterations and conditional processing.
4. A flexible template processing mechanism which operates within an object oriented environment.
5. A flexible and easily extended formatting mechanism for output.
6. A mechanism for template creation and editing.
7. Special support for complex object presentation.
8. Templates that support parsing of the input file and mapping of its contents into a complex object structure suitable for update processing.
9. A mechanism to extract data from the input stream.
10. A mechanism to map the structure of the input file into a complex object structure suitable for internal processing.
11. A mechanism to organize the updating data from the input file and to apply it within the application.
12. A flexible and easily extended formatting mechanism.
13. A mechanism for input template creation and editing.

Figure 2

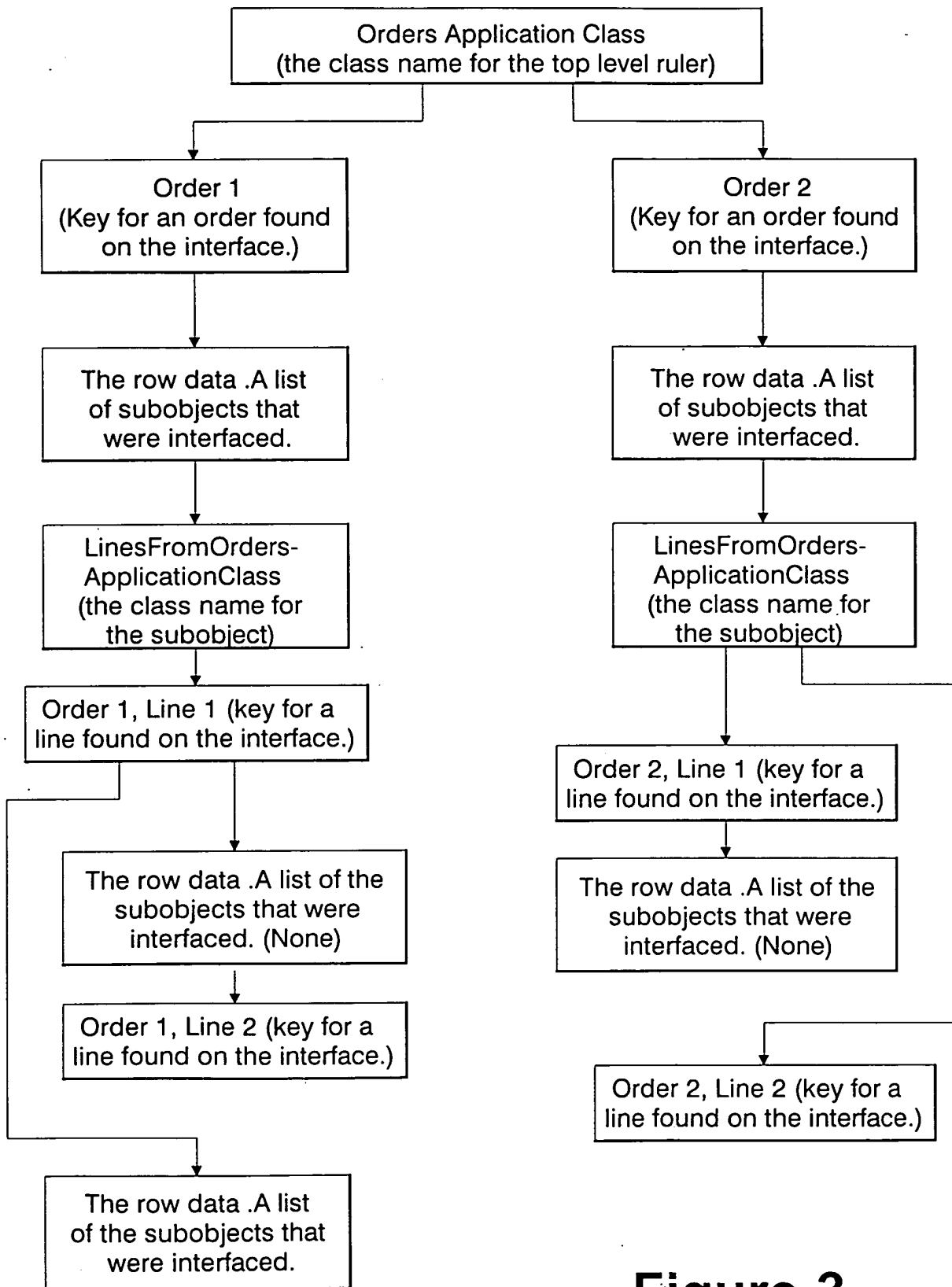


Figure 3